

Amendments To The Claims:

This Listing Of Claims Will Replace All Prior Versions, And Listings, Of Claims In The Application:

Listing Of Claims:

Claim 1 (previously amended): A method of receiving data from a network, comprising:
issuing a request directing a transfer of data from one of a plurality of device ports to a storage unit and specifying a thread from among a plurality of processing program threads to process the data.

Claim 2 (previously amended): The method of claim 1, further comprising:
determining if at least one of the plurality of device ports coupled to the network require service.

Claim 3 (previously amended): The method of claim 2, further comprising:
transferring the data to the storage unit and signaling to the specified program thread that the data is ready for processing.

Claim 4 (original): The method of claim 2, wherein determining comprises:
interrogating the plurality of device ports to identify which of the plurality of device ports require service.

Claim 5 (original): The method of claim 4, wherein determining further comprises:
preparing control information corresponding to those device ports identified as requiring service.

Claim 6 (previously amended): The method of claim 5, wherein the control information comprises indicators each associated with a device port receive FIFO in a corresponding one of the device ports.

Claim 7 (currently amended): The method of claim 6, wherein interrogating comprises: polling the state of ~~the~~ ready flags to determine if the indicators are asserted, the assertion of the indicators indicating that the corresponding device ports have data ready for transfer.

Claim 8 (previously amended): The method of claim 7, wherein the indicators indicate that the associated device port receive FIFO has reached a threshold level of fullness.

C
Claim 9 (previously amended): The method of claim 8, wherein the indicators indicate that the associated device port receive FIFO stores a full network packet.

Claim 10 (previously amended): The method of claim 5, further comprising: maintaining a receive ready count, the receive ready count being incremented in response to the control information being prepared.

Claim 11 (original): The method of claim 5, wherein preparing control information further comprises:

writing a flag to a control and status register for each device port in the plurality of device ports that is determined to require service.

Claim 12 (original): The method of claim 11, wherein issuing comprises: obtaining the control information from the control and status register; and selecting from each device port in the plurality of device ports having set bits in the control and status register a port for servicing.

Claim 13 (currently amended): The method of claim 12 ~~1~~, ~~wherein issuing~~ further ~~comprises~~ comprising:

determining which among the plurality of program threads is available; and
assigning an available program thread ~~to the selected port~~ to process the data.

Claim 14 (original): The method of claim 12, wherein selecting a port comprises:
using the receive ready count to determine if the ready flags reflect current status of the
device port.

Claim 15 (previously amended): The method of claim 3, further comprising:
maintaining a receive request count for counting transfer of data to the storage unit, the
receive request count being incremented by one upon the transfer of the data to the storage unit
and signaling to the specified program thread.

Claim 16 (previously amended): The method of claim 15, wherein selecting a port
further comprises:
using the receive request count to determine if the indicators reflect current status of the
device ports.

Claim 17 (previously amended): A method of receiving data from a plurality of
peripheral ports, comprising:
determining that the one of the plurality of peripheral ports requires servicing;
issuing a receive request based on the determination, the receive request directing the
transfer of data from the one of the plurality of peripheral ports to a buffer memory and
specifying a program thread from among a plurality of processing program threads to process
the data; and
transferring the data to the buffer memory and signaling to the specified thread that the
data is ready for processing.

Claim 18 (currently amended): An article comprising a computer-readable medium ~~which~~ that stores computer-executable instructions for receiving data from a plurality of ports, the instructions causing a computer to:

issue a receive request directing a transfer of data from one of a plurality of device ports to a storage buffer and specifying a program thread from among a plurality of processing program threads to process the data.

Claim 19 (previously amended): The article of claim 18, the article further comprises instructions causing a computer to:

determine if at least one of the plurality of device ports coupled to the network require service.

Claim 20 (previously amended): The article of claim 19, the article further comprises instructions causing a computer to:

transfer the data to the storage unit and signal to the specified program thread that the data is ready for processing.

Claim 21 (original): The article of claim 19, wherein the instructions to determine comprise instructions causing a computer to:

interrogate the plurality of device ports to identify which of the plurality of device ports require service; and

prepare control information corresponding to those device ports identified as requiring service.

Claim 22 (previously amended): The article of claim 21, the article further comprises instructions causing a computer to:

maintain a receive ready count, the receive ready count being incremented in response to the control information being prepared.

Claim 23 (original): The article of claim 22, wherein the instructions to issue comprise instructions causing a computer to:

use the receive ready count to check the current status of the device port.

Claim 24 (previously amended): The article of 19, the article further comprises instructions causing a computer to:

maintain a receive request count for counting transfer of data to the storage unit, the receive request count being incremented by one upon the transfer of the data to the buffer memory and signaling to the specified program thread.

Claim 25 (original): The article of claim 24, wherein the instructions to issue comprise instructions causing a computer to:

use the receive request count to check the current status of the device ports.

Claim 26 (withdrawn).

Claim 27 (withdrawn).

Claim 28 (withdrawn).

Claim 29 (new). A computer program, disposed on a computer readable medium, the program including instructions to cause a processor to:

determine availability of threads provided by a set of independently programmable multi-threaded engines, the threads provided by the set of independently programmable multi-threaded engines having associated program counters;

select at least one thread from the set of threads based, at least in part, on the availability of threads in the set of threads; and

assign a received packet to the at least one thread selected from the set of threads.

Claim 30 (new). The computer program of claim 29 wherein the program instructions, when executed, form instructions of at least one thread of at least one engine of the set of independently programmable multi-threaded engines.

Claim 31 (new). The computer program of claim 29 wherein the packet comprises a packet received from a media access controller.

Claim 32 (new). The computer program of claim 31 wherein the packet comprises a packet divided into multiple packet segments by the media access controller.

Claim 33 (new). The computer program of claim 32 wherein the instructions to assign comprise instructions to:

store a thread identifier with at least one of the segments of the received packet.

Claim 34 (new). The computer program of claim 32 wherein the segments of the packet comprise segments associated with at least one of: a start of the packet identifier, an end of the packet identifier, and a middle of the packet identifier.

Claim 35 (new). The computer program of claim 29 wherein the instructions to select comprise instructions to:

select at least one thread, based at least in part, on a static assignment of ports to threads.

Claim 36 (new). The computer program of claim 29 wherein the instructions to select comprise instructions to:

select at least one thread, based at least in part, on capabilities of the threads.

Claim 37 (new). The computer program of claim 36, wherein the capabilities of the threads comprise network protocol operations provided by the threads.

Claim 38 (new). The computer program of claim 29, wherein the multiple engines comprises multiple engines integrated within a processor.

Claim 39 (new). A system comprising:
at least one media access controller;
at least one processor, at least one of the at least one processor comprising:
multiple independently programmable engines that collectively provide a set of multiple threads having associated program counters;
at least one interface to the at least one media access controller; and
a computer program, disposed on a computer readable medium, the program including instructions to cause the processor to:
determine availability of threads provided by the set of independently programmable engines;
select at least one thread from the set of threads based, at least in part, on the availability of threads in the set of threads; and
assign a received packet to the at least one thread selected from the set of threads.

Claim 40 (new). The system of claim 39 wherein the multiple engines comprise multiple multi-threaded engines.

Claim 41 (new). The system of claim 40 wherein the program instructions, when executed, form at least one thread of at least one engine of the set of independently programmable engines.

Claim 42 (new). The system of claim 39 wherein the instructions to select comprise instructions to select based, at least in part, on capabilities of the threads.

Claim 43 (new). The system of claim 39 wherein the at least one media access controller comprises at least one Ethernet media access controller.